

(51)Int.Cl.<sup>5</sup>  
G 0 6 F 9/42  
9/38

識別記号  
3 2 0 A 9189-5B  
3 3 0 F 9290-5B

庁内整理番号  
F I

技術表示箇所

審査請求 未請求 請求項の数 4 (全 13 頁)

(21)出願番号	特願平3-281030	(71)出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目 6 番地
(22)出願日	平成 3 年(1991)10月28日	(72)発明者	成田 進 東京都国分寺市東恋ヶ窪 1 丁目280番地 株式会社日立製作所中央研究所内
		(72)発明者	荒川 文男 東京都国分寺市東恋ヶ窪 1 丁目280番地 株式会社日立製作所中央研究所内
		(72)発明者	内山 邦男 東京都国分寺市東恋ヶ窪 1 丁目280番地 株式会社日立製作所中央研究所内
		(74)代理人	弁理士 小川 勝男
		最終頁に続く	

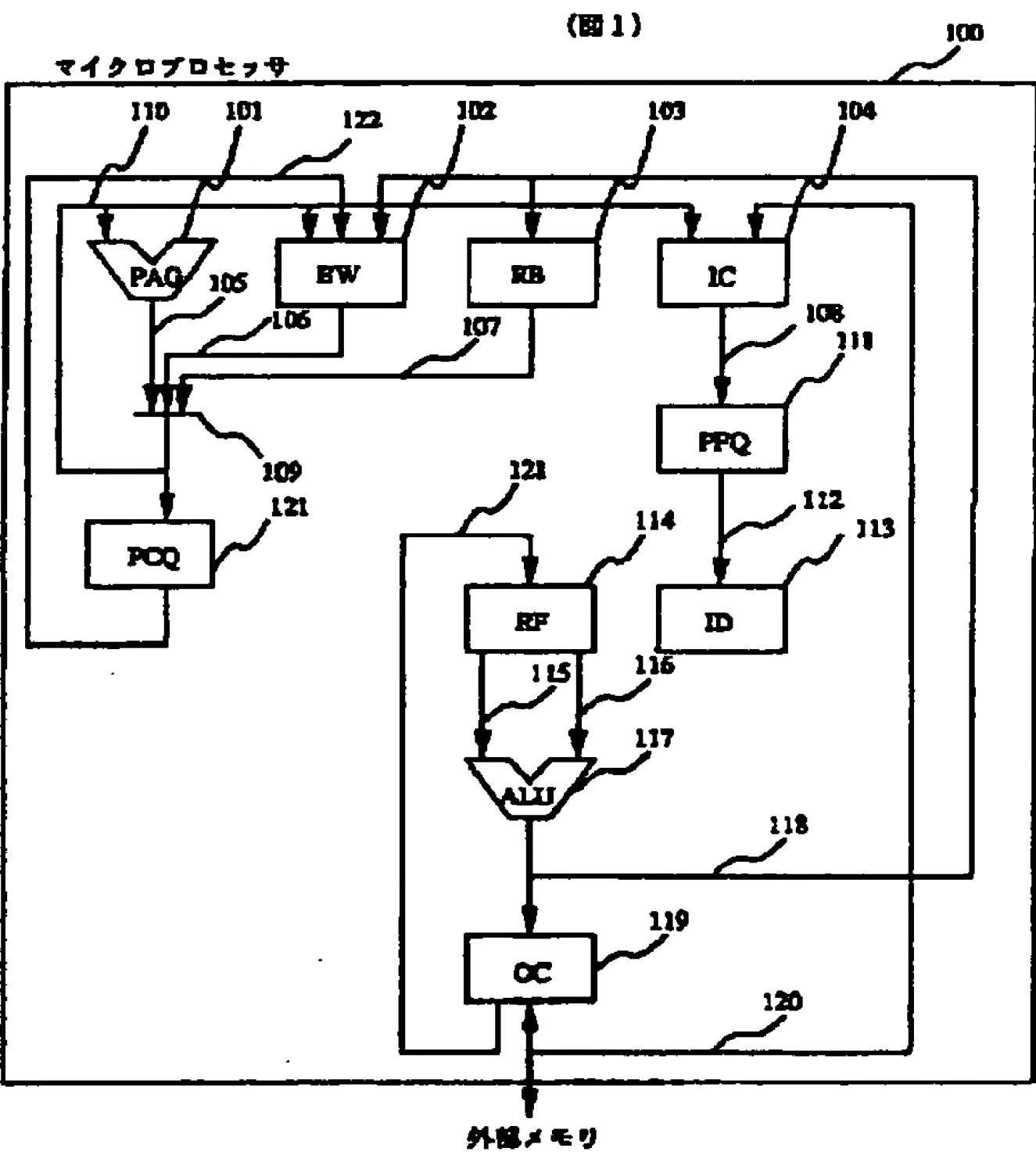
(54)【発明の名称】 分岐命令バッファを有するデータプロセッサ

(57)【要約】

【目的】マイクロプロセッサの分岐処理を高速化する。特に、リターン命令を含む無条件分岐命令を高速化する。

【構成】分岐命令アドレスと分岐先アドレス、さらに分岐命令タイプを組にして分岐履歴情報として分岐命令バッファBW102に保持する。また、リターンアドレスを保持するリターンバッファRB103を設ける。BW103の検索はプリフェッチアドレスを用いて行い、ヒット時には分岐命令タイプを用いて、BW102とRB103のいずれかの分岐先アドレスを選択し、プリフェッチアドレスを切り替えることで先行分岐する。

【効果】分岐時のオーバーヘッドを零にできる。リターン命令を含むあらゆる分岐命令に適用可能である。



## 【特許請求の範囲】

【請求項1】分岐命令のアドレスと分岐先アドレスとを格納する第1のバッファと、サブルーチンからのリターンアドレスを格納する第2のバッファとを具備してなり、命令をプリフェッチする際に、プリフェッチされる命令のアドレスと上記第1のバッファ中に格納された上記分岐命令のアドレスを比較し、上記アドレスの比較が一致した際、分岐命令の種類がサブルーチンからのリターン命令以外であった場合には上記第1のバッファから上記分岐先アドレスを読み出すことによって次命令のプリフェッチアドレスを生成し、分岐命令の種類がリターン命令であった場合には上記第2のバッファから上記リターンアドレスを読み出すことによって次命令のプリフェッチアドレスを生成することを特徴とするデータプロセッサ。

【請求項2】上記第1のバッファと上記第2のバッファとはマイクロプロセッサの半導体チップに構成されてなることを特徴とする請求項1に記載のデータプロセッサ。

【請求項3】上記第2のバッファはラストイン・ファーストアウトのバッファであることを特徴とする請求項1および請求項2のいずれかに記載のデータプロセッサ。

【請求項4】上記第1のバッファは上記分岐命令の種類を示す情報を、上記分岐命令のアドレスと上記分岐先アドレスと組で格納することを特徴とする請求項1から請求項3のいずれかに記載のデータプロセッサ。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は分岐命令を実行する機能と命令をプリフェッチする機能を共に有するデータプロセッサに関わり、特に分岐履歴情報に対して命令プリフェッチ機能を連動させることにより、高速に分岐処理を実行するデータプロセッサに関する。

## 【0002】

【従来の技術】従来、分岐処理の高速化のために、分岐の履歴情報として分岐命令のアドレスと分岐先命令のアドレスをバッファに格納しておき、命令プリフェッチ時にプリフェッチアドレスを入力として履歴情報を検索し、先行的に分岐するデータ装置について、特開平1-240931号公報に記載がある。

【0003】また従来、同じく分岐処理の高速化のために、分岐の履歴情報として分岐命令に先行する命令のアドレスと分岐先命令のアドレスをバッファに格納しておき、命令解読時に命令アドレスを入力として履歴情報を検索し、無条件分岐命令の実行処理を削除することで高速に分岐するデータプロセッサについて、特開平2-166520号公報に記載がある。

## 【0004】

【発明が解決しようとする課題】上記従来技術のいずれ

も、主に無条件分岐命令の分岐処理を高速化するものであるが、無条件分岐命令の一つであるサブルーチンからのリターン命令（rts命令）の分岐処理には対応できないことが本願発明者等の検討により明らかとされた。これは以下のような理由による。

【0005】上記従来技術はいずれも、分岐命令の各々は常に分岐先アドレスが不変であるという仮定に立って、分岐の履歴情報が次の分岐処理にも役に立つと考えている。しかしながら、サブルーチンからのリターン命令の場合にはその仮定が成り立たない。なぜならば、リターン命令はサブルーチンからコール元のルーチンへのリターンに用いられる命令であるため、コール元が異なれば当然、リターン先のアドレスも異なるものとなるからである。

【0006】サブルーチンのコールとリターンの処理は、基本的には次のように行う。

【0007】まず、コール元のルーチンではサブルーチンコール命令を実行する。サブルーチンコール命令（bsr命令）は、まずリターン先アドレスを計算し、スタックと呼ばれるメモリ上のLIFOキュー（ソフトウェアで作成する）に、リターンアドレスをストアする。そして、そのサブルーチンコール命令（bsr命令）によって、サブルーチンへと分岐する。続いてサブルーチンでの処理を実行し、最後にリターン命令を実行する。リターン命令はまずスタックからリターンアドレスを読み出し、続いてそのリターンアドレスへと分岐することでコール元のルーチンにリターンする。

【0008】このように、リターンアドレスはサブルーチンコール命令（bsr命令）のアドレスによって定まるものであって、一つのサブルーチンをコールするサブルーチンコール命令（bsr命令）が複数ある場合には、リターン命令（rts命令）に対応したリターンアドレスは一意には定まらない。

【0009】従って、本発明の目的とするところは、高速に分岐処理を実行することが可能であるとともに、無条件分岐命令の一つであるリターン命令の分岐処理に対応することの可能なデータプロセッサを提供することにある。

## 【0010】

【課題を解決するための手段】上記目的を解決するためには、次の二つの手段が採用される。

【0011】（1）第1のバッファに、データプロセッサが一度実行した分岐命令のアドレス、分岐先アドレス、分岐命令の種類を示す分岐履歴情報を格納する。

【0012】（2）第2のバッファにサブルーチンからのリターンアドレスを格納する。

## 【0013】

【作用】第1のバッファには、データプロセッサが一度実行した分岐命令のアドレス、分岐先アドレス、分岐命令の種類を示す分岐履歴情報が格納されている。従っ

て、同一の分岐命令をデータプロセッサが実行するに際して、第1のバッファから分岐先アドレスが高速に読み出され、次命令プリフェッチアドレスが生成されることができる。

【0014】一方、リターンアドレスを格納する第2のバッファは、サブルーチンコール命令（b s r命令）を実行する際に書き込みを行い、リターン命令（r t s命令）がプリフェッチされたときに読み出しを行うL I F O（ラストイン・ファーストアウト）キューである。尚、リターン命令（r t s命令）を実行するまで、第2のバッファのポインタの更新は行わない。言い替えると、第2のバッファはメモリ上にあるスタックの一部（リターンアドレス）のコピーを格納するキャッシュメモリである。

【0015】リターン命令がプリフェッチされたことの検出は、分岐履歴情報を格納する第1のバッファを用いて行う。すなわち、リターン命令（r t s命令）もまた他の分岐命令（b r a命令、b s r命令）と同様に、分岐履歴情報をこの第1のバッファに登録する。但し、リターン命令の場合はその検出にのみ第1のバッファを使用し、分岐先アドレスのフィールドは使用しない。代りに、上記のリターンアドレスを格納する第2のバッファから、分岐先アドレスを得る。

【0016】リターン命令を他の分岐命令と共に、同一の分岐履歴情報を格納する第1のバッファに登録するため、分岐先アドレスとしていずれのバッファのそれを用いるかを判断するための情報が必要になる。この問題を解決するため、分岐履歴情報を格納する第1のバッファ内に、分岐命令の種類を示す情報（分岐命令タイプ）を格納している。

【0017】また別の解決方法として、単純に第1と第2のバッファのいずれの分岐先アドレスを使用するかを示す情報（選択候補が二つの場合には1ビット）を格納してもいい。

【0018】また第1と第2のいずれのバッファの分岐先アドレスを用いるかを示す情報は、必ずしも分岐命令履歴情報を格納する第1バッファ内に持つ必要はない。データプロセッサの制御回路内に、同等の情報を格納することでも、同じ効果を期待できる。

【0019】また分岐履歴情報の検索に、プリフェッチアドレスではなく命令アドレスを使用することも可能である。但しこの方法では、以下の実施例に示すほどの分岐高速化の効果は期待できない。

【0020】また分岐履歴情報の検索に特開平2-166520号公報のように、分岐命令に先行する命令のアドレスを使用することも可能である。マイクロプロセッサのアーキテクチャによっては、一命令以上の先行を有する場合もある。

【0021】

【実施例】図1は本発明の一実施例であるマイクロプロ

セッサのブロック図である。本発明は命令プリフェッチ時の分岐処理を高速化する手法であるため、ここでは命令プリフェッチ部を中心に説明する。

#### 【0022】1. マイクロプロセッサの内部構造

図1を用いてマイクロプロセッサの内部構造を説明する。図1のマイクロプロセッサの各構成要素の機能は次のとおりである。

【0023】PAG 101 プリフェッチアドレス発生器（加算器）。

BW 102 分岐命令用バッファ。分岐先アドレスを保持する。

RB 103 リターンバッファ。リターンアドレスを保持する。

IC 104 命令キャッシュ。

PFQ 111 プリフェッチキュー。

PCQ 121 命令アドレスキュー。PAGで生成した命令アドレスを保持する。

ID 113 命令デコーダ。

RF 114 レジスタファイル。

ALU 117 整数論理演算器。

OC 119 オペランドキャッシュ。

上記構成要素のうち、PAG、BW、RB、IC、PFQ、PCQが命令プリフェッチ部に含まれる。以下順に各構成要素を説明する。

【0024】29ビット加算器であるPAG101は、プリフェッチアドレスを生成する。プログラムを逐次実行する場合、すなわち分岐時以外には、一回プリフェッチする毎にプリフェッチアドレスに固定値を加算して次のプリフェッチアドレスを生成する。加算する固定値は一度にプリフェッチする命令のバイト幅と等しく、外部メモリとIC（命令キャッシュ）間のデータ線幅が8バイトの場合、加算値は8である。加算器への入力の方は信号線110であり、これにはプリフェッチアドレスの値が出力されている。今一方の値は固定値であり、本実施例では値8（LSBへのキャリー1ビットで値8を表現できる）である。これらの加算結果は信号線105に出力される。

【0025】分岐命令用バッファBW102は、分岐命令の履歴すなわち分岐命令のアドレスと分岐先アドレス、さらに分岐命令タイプを一組にして履歴情報として記憶しておく。命令プリフェッチ時にはプリフェッチアドレスと履歴情報中の分岐命令アドレスを比較し、一致した場合には分岐先アドレスと分岐命令タイプを出力する。

【0026】リターンバッファRB103は、分岐命令の一つであるリターン命令用にリターンアドレスを保持するL I F O（ラストイン・ファーストアウト）キューである。

【0027】命令キャッシュIC104は、信号線110を通してプリフェッチアドレス（29ビット）を入力

し、対応する命令（64ビット）をキャッシュから読み出し、信号線108に出力する。入力したアドレスに対応する命令がキャッシュIC104内にはない場合には、外部メモリアクセスを起動し、信号線120（64ビット）を通して命令が外部メモリから読み出されたキャッシュIC104に書き込まれる。

【0028】プリフェッチキューPFQ111はプリフェッチした命令を保持するFIFO（ファーストイン・ファーストアウト）キューの機能と、命令を整列する機能（本実施例では64ビットから16ビットに整列する）を持つ。入力は信号線108（64ビット幅）、出力は信号線112（16ビット幅）である。

【0029】命令アドレスキューPCQ121は、PAG101で生成したプリフェッチアドレスを保持するFIFOキューである。入力は3入力セレクタ109の出力信号110（29ビット）であり、出力は信号線122（29ビット）である。

【0030】命令デコーダID113には、PFQ111から信号線112（16ビット）を通して命令が入力される。入力される各命令の長さは16ビット単位で整列された状態であり、命令のデコード結果は制御線を通して各構成要素に接続される。ただし図1では制御信号線を省略してある。

【0031】レジスタファイルRF114は、本実施例では32ビット幅のレジスタ16本から構成され、入力ポートを一つ、出力ポートを二つ持っている。かつこれらの三つのポートは同時に動作可能である。各ポートのビット幅は32ビットであり、それぞれ信号線121、115、116に接続されている。

【0032】32ビット幅の整数論理演算器ALU117の入力はレジスタファイルから出力された信号線115、116であり、演算結果を信号線118に出力する。本実施例ではデータの演算とアドレスの計算を共にALU117で処理する。そのため、信号線118にはデータまたはアドレスの何れかが出力される。

【0033】オペランドキャッシュOC119は、オペランドをフェッチする場合には、信号線118にアドレスを入力としてアクセスされ、結果として得られたデータを信号線121に出力してレジスタファイル114に値を転送する。オペランドストアの場合には、まず最初のサイクルでストアアドレスをALU117からOC119に信号線118を用いて転送し、そのアドレスはOC119内に保持する。続く次のサイクルでは信号線118を用いてストアデータを転送し、OC119及び外部メモリへのオペランドストア処理を行う。また、オペランドフェッチの際にアクセスしたデータがオペランドキャッシュ119内に無い場合には、外部メモリアクセスを起動し、外部メモリからOC119に信号線120を通してオペランドの転送を行う。

【0034】2. パイプライン処理の流れ

図2から図6を用いて、図1の本実施例のマイクロプロセッサのパイプライン処理の流れを説明する。

【0035】2. 1 分岐の無い時のパイプラインの流れ

図2は分岐が無い時と時のパイプラインの処理の流れを示している。横軸は時間であり、 $t_0$ 、 $t_1$ 、...のそれぞれが1サイクルである。縦軸にはパイプラインの各ステージの処理をとっている。

【0036】ステージiは命令プリフェッチステージであり、図1のPAG101、BW102、RB103、IC104、PFQ111、PCQ121の処理が含まれる。例えば図2において、時刻 $t_0$ では命令1のプリフェッチが行われ、PFQ111から信号線112を通して命令1が次の処理ステージ（命令デコード）に転送される。尚、図2中の命令読み出しはすべて命令キャッシュIC104から行われると仮定している。

【0037】ステージdは命令デコードステージであり、図1のID113、RF114の処理が含まれる。図2の時刻 $t_1$ において命令1は命令デコーダID113でデコードされ、続いてデコード結果に基づいてレジスタファイルRF114からの読み出しを行う。読み出されたデータは信号線115、116に出力される。

【0038】ステージeは演算及びアドレス計算ステージであり、図1のALU117の処理が含まれる。図2の時刻 $t_2$ において命令1は命令デコーダID113からの制御に基づいて、ALU117を用いて演算を実行する。入力は信号線115、116であり、結果を信号線118に出力する。

【0039】ステージaはオペランドアクセスステージであり、図1のOC119の処理が含まれる。図2の時刻 $t_3$ において命令1は命令デコーダ113からの制御に基づいて、オペランドアクセス処理を実行する。実行する処理は次の3種類である。

【0040】（1）オペランドフェッチ（OC119内にフェッチすべきデータが無い場合には、外部メモリからOC119へデータを転送後、再度OC119からフェッチ処理を実行する。）

（2）オペランドストア（OC119と外部メモリの両方にデータをストアする。）

（3）データ転送（ALU117の演算結果をRF114にストアする場合の処理。）

いずれも入力は信号線118であり、上記処理（1）と（3）における出力は信号線121に行う。また、外部メモリとのデータの入出力には信号線120を用いる。外部メモリへのアドレス出力線は図1では省略している。

【0041】ステージsはレジスタストアステージであり、図1のRF114の処理が含まれる。図2の時刻 $t_4$ において命令1は命令デコーダ113からの制御に基づいて、レジスタストアを実行する。入力は信号線12



1である。

#### 【0042】 2. 2 分岐時のパイプラインの流れ

図3は無条件分岐命令 *bra* の実行時のパイプラインの流れを示している。実行している命令の種類が無条件分岐命令であることは、この命令の処理が命令デコードステージ *d* を終了した時点すなわち時刻 *t* 0 で判る。但し分岐先アドレスはアドレス計算ステージ *e* が終了する時点すなわち時刻 *t* 1 で明らかになる。そのため、分岐先アドレスの命令 11、12の命令プリフェッチは時刻 *t* 2で行われる。そしてその結果として、1回の分岐あたり2サイクルのオーバーヘッド（パイプライン処理のあき時間）が生じる。

【0043】図4は、図1の分岐命令用バッファ BW102を用いて *bra* 命令の分岐処理を高速化した場合のパイプラインの流れを示している。図3と比べて、オーバーヘッドが0サイクルになっている。これは時刻 *t* 0の *i* ステージにおいて、*bra* 命令がプリフェッチされた事をBW102の履歴情報を用いて検出し、時刻 *t* 1では既に分岐先命令である命令 11をプリフェッチできている結果である。また *bra* 命令自体の処理は分岐処理以外何も無いので、*i* ステージにおいて *bra* 命令そのものが削除され、*d* ステージには *bra* 命令は転送されていない。

【0044】図5は図4と同様に、図1の分岐命令バッファ BW102を用いて分岐処理を高速化した例である。但し分岐命令は、サブルーチンコール命令 *bsr* である。サブルーチンコールのための *bsr* 命令は無条件分岐のための *bra* 命令と異なり、分岐以外にも実行すべき処理があるため、*bra* 命令のように削除する訳には行かない。そのため分岐には2サイクルの実行時間を必要とする。分岐によるオーバーヘッドは図4の場合と同じく0サイクルである。

【0045】図6はやはり図4と同様に、分岐命令バッファ BW102を用いて分岐処理を高速化した例である。但し分岐命令はリターン命令 *rts* である。*rts* 命令は *bsr* 命令と同じ理由で、削除することはできない。そのため分岐には1サイクルを要する。但し分岐命令バッファ BW102の効果によってオーバーヘッドは0サイクルである。

【0046】以上説明したように、無条件分岐命令 *bra* とサブルーチンコール命令 *bsr* の処理には分岐命令用バッファ BW102を用い、リターンバッファ RB103は用いない。しかし、リターン命令 *rts* の処理にはBW102とRB103の両方を用いる。動作フローを説明する前に、次にBWとRBの構造を説明する。

### 3. BWとRBの構造と動作

#### 3. 1 BWの構造と動作

図7は、図1の分岐命令バッファ BW102をより詳細に説明する構成図である。分岐命令バッファ BWはアドレスデコーダ 201、アドレスタグ部 BWA202、デ

ータ部 BWD203、及び一致比較器 CMP209で構成されている。ここにアドレスデコーダ 201は5ビットデコーダであり、デコードの結果としてBWA202とBWD203の32個のエントリの一つを指定するポインタを得る。BWA202とBWD203とは、いずれもRAMメモリを用いて構成されているが、連想メモリを用いて構成することも可能である。エントリすなわちワード数は共に32であり、ビット幅はそれぞれ32、33ビットである。

【0047】32ビットの内訳は、BWA202が分岐命令アドレス31ビット（入力線122、出力線206、213）、バリッドビット1ビット（入力線204、出力線207）である。またBWD203では分岐先アドレス31ビット（入力線118、出力線106）、分岐命令タイプ2ビット（入力線205、出力線208）である。分岐命令タイプは、前述の無条件分岐命令 *bra*、サブルーチンコール命令 *bsr*、リターン命令 *rts* の命令の種別を区別する情報である。CMP209は24ビット幅の一致比較器であり、一致比較結果を示す信号 210（一致時には値1、不一致時には値0となる）はAND回路 211でバリッド信号 207と論理積が取られ、AND回路 211の出力はヒット信号 212となって命令プリフェッチ部内のBW制御回路へと出力される。

【0048】分岐命令バッファ 102 BWの書き込み動作は、次の通りである。

【0049】まず29ビットのプリフェッチアドレス信号線 110の下位5ビットが、アドレスデコーダ 201に入力される。続いて、アドレスデコーダ 201でアドレスデコードが行われ、32個のエントリのうちのひとつが選択される。この時、BWA202とBWD203に入力されている信号線 122、204、118、205の値が、選択されたエントリに同時に書き込まれる。分岐命令アドレス 122は図1の命令アドレスキュー PCQ121からの出力信号線、バリッドビット 204は命令プリフェッチ部の制御回路からの出力、分岐先アドレス 118は図1のALU117からの出力信号線（アドレス計算の結果）、分岐命令タイプは命令デコーダ 1D113の出力情報を命令プリフェッチ部の制御回路でタイミング調整した信号である。

【0050】分岐命令バッファ BWの読み出し動作は、次の通りである。

【0051】書き込み動作と同様に、29ビットのプリフェッチアドレス信号線 110の下位5ビットがアドレスデコーダ 201に、残りの24ビットが一致比較器 CMP209に入力される。続いてアドレスデコーダ 201でアドレスデコードが行われ、32個のエントリのうちのひとつが選択される。BWAとBWDの選択されたエントリのデータが読み出され、信号線 206、213、207、106、208にそれぞれ出力される。

【0052】信号線206に出力された分岐命令アドレスは、選択されたエントリに登録されている分岐命令のアドレスの上位24ビットである。エントリの選択にはアドレスの下位5ビットのみを用いているので、登録されている分岐命令のアドレスとプリフェッチ中の命令に含まれる分岐命令のアドレスが同一であるかどうかを調べるために、CMP209を用いてアドレスの上位24ビットの一致比較を行う。

【0053】この24ビットに続くアドレスの下位7ビットが、信号線213に出力される。信号線213の上位5ビットは、プリフェッチアドレス110の5ビットと一致している。それ故に、この5ビットをBWAから削除することもまた可能である。信号線213の下位2ビットはプリフェッチされた命令列8バイト中での分岐命令(2バイト長)の位置を示しており、プリフェッチキューPFQ121の制御情報として使用される。

【0054】また、バリッドビット207は読み出されたエントリの情報が有効であるか否かを示している。このバリッドビット207は読み出されたエントリの情報が有効な時は"1"となり、無効な時は"0"となる。そこでバリッド信号207と一致比較結果の信号210の論理積をとることで、BWA202とBWD203とから読み出したデータがプリフェッチアドレス110に対応した有効な情報であるか否かを示すヒット信号212を生成できる。

【0055】BWD203から読み出された分岐先アドレス106は、命令プリフェッチ部内での分岐処理に用いられる。すなわち、図1においてBW102から出力された分岐先アドレス106はセクタ109、信号線110を通して、プリフェッチアドレスとして命令キャッシュIC104、分岐命令用バッファBW102、プリフェッチアドレス発生器PAG101に入力される。

【0056】また、分岐命令タイプ208は、読み出されたエントリの分岐命令が無条件分岐命令bra、サブルーチンコール命令bsr、リターン命令rtsのいずれであるかを示している。命令プリフェッチ部内の制御回路はこの情報を受けて、図4から図6に示した処理のいずれかを実行する。

### 【0057】3. 2 リターンバッファRBの構造と動作

図8は、図1のリターンバッファRB103の構成をより詳細に示している。RBは、デコーダ302とRAMメモリRBD303とで構成される。デコーダ302は4ビットデコーダであり、4ビット幅のポインタ301を入力として、RBD303の16エントリのうちの一つを選択する。RBD303は16エントリ、ビット幅32ビットのRAMである。32ビットの内訳は、リターンアドレス31ビット(入力線118、出力線107)、バリッドビット1ビット(入力線304、出力線305)である。RB303では読み出されたバリッド

ビットがそのままヒット信号となる。

【0058】リターンバッファRB103への書き込み動作は、次の通りである。

【0059】4ビットのポインタ301は、命令プリフェッチ部の制御回路から与えられる。ポインタ301はデコーダ302でデコードされ、RBD303のエントリの一つを選択する。この時、RBD303に入力されているリターンアドレス118とバリッド信号304の値とが選択されたエントリに書き込まれる。バリッド信号304は命令プリフェッチ部の制御回路から与えられる。RB103への書き込みは、サブルーチンコール命令(例えばbsr命令)の実行時に起動される。そしてその際、ポインタを一つ進める。

【0060】リターンバッファRB103からの読み出し動作は、次の通りである。

【0061】書き込み時と同じく、4ビットのポインタ301は命令プリフェッチ部の制御回路から与えられる。ポインタ301はデコーダ302でデコードされ、RBD303のエントリの一つを選択する。選択されたエントリのデータは、信号線107と信号線305に同時に出力される。BW102で読み出された分岐命令のタイプがリターン命令rtsでありかつリターンバッファRB103から読み出されたヒット(バリッド)信号が値1であった場合、リターンアドレス107を次のプリフェッチアドレスとして、図1のセクタ109、信号線110を通して命令キャッシュIC104、分岐命令用バッファBW102、プリフェッチアドレス発生器PAG101に転送する。リターンバッファRB103のポインタは、サブルーチンリターン命令(例えばrts命令)の実行時に一つ戻す。

【0062】サブルーチンリターン命令rtsが無条件分岐命令bra、サブルーチンコール命令bsrと異なり、分岐先アドレスを分岐命令用バッファBW102内に保持できない理由は次のとおりである。

【0063】リターン命令rtsは、サブルーチンからのリターンのために使用される。すなわち、サブルーチンの最後の命令として、サブルーチンをコールしたルーチンのコール命令の次の命令に分岐する。一方、リターンアドレスはコール時にスタックに退避され、リターン時にはスタックから回復される。そのため、どのルーチンからコールされるかによって、同じサブルーチンリターン命令rtsであっても、リターン先アドレスが異なる。

【0064】これに対して分岐命令バッファBW102では、分岐命令のアドレスと分岐先アドレスを一組にした履歴情報を保持する。前述のようにサブルーチンリターン命令rtsの場合には分岐命令アドレスと分岐先アドレスの関係が一意ではないから、サブルーチンリターン命令rtsの分岐先リターンアドレスを分岐命令用バッファBW102内に保持することはできない。そこ

で、スタックのリターンアドレスのコピーを保持するバッファであるRB103を設け、このバッファRB103からリターンアドレスを得ることでこの問題を解決している。

#### 【0065】4. 動作フロー

図9、図10は上記に説明した本実施例によるデータプロセッサの命令プリフェッチ部の制御基本フローである。分岐、リセット、例外発生時の状態遷移は省略した。以下、図9、図10を用いて制御フローを説明する。

【0066】ステップ900はプリフェッチキューPFQ111のあきを待っている状態である。PFQ111はFIFOキューであるので、データが満杯になった場合には、命令デコーダID113へのデータ転送によって空きができるまで、次のデータ書き込みができなくなる。そのためステップ901の判定動作（PFQが満杯か否か）によって、再び待ち状態ステップ900に遷移するか、それともステップ902に遷移するかが決定される。

【0067】ステップ902は、命令キャッシュIC104、分岐命令バッファBW102及びリターンバッファRB103の読み出しを行う状態である。それらの読み出しの結果をステップ903から906で判定し、次の動作状態を907から910のどれか一つに決定する。

【0068】ステップ903は命令キャッシュIC104の読み出しが成功（ヒット信号の値が1）したか否かを判定し、成功した場合にはステップ904へ、失敗した場合にはステップ910に遷移する。

【0069】ステップ904は分岐命令バッファBW102の読み出しが成功したか否か、すなわち図7のヒット信号212の値が1であるか否かを判定する。読み出しが成功した場合にはステップ905へ、失敗した場合にはステップ909へ遷移する。

【0070】ステップ905は分岐命令バッファBW102から読み出された分岐命令タイプ情報（図7の信号208）がリターン命令rtsを指示するものか否かを判定する。リターン命令rtsの場合にはステップ906へ、そうでない場合にはステップ907へ遷移する。

【0071】ステップ906はリターンバッファRB103の読み出しが成功したか否か、すなわち図8のヒット信号305の値が1であるか否かを判定する。成功した場合にはステップ908へ、失敗した場合にはステップ909へ遷移する。

【0072】ステップ907は分岐命令バッファBW102にヒットし、かつその分岐命令が無条件分岐命令braまたはサブルーチンコール命令bsrである場合に遷移する状態であり、次のサイクルのプリフェッチアドレスとしてBW102から出力された分岐先アドレス信号106を用いる。具体的には、図1のセクタ109

により信号106を選択し、その値を信号110に出力する。

【0073】ステップ908は分岐命令バッファBW102とリターンバッファRB103の両方にヒットし、かつその分岐命令がリターン命令rtsである場合に遷移する状態であり、次のサイクルのプリフェッチアドレスとしてRB103から出力されたリターンアドレス信号107を用いる。具体的には、図1のセクタ109により信号107を選択し、その値を信号110に出力する。

【0074】ステップ909は命令キャッシュIC104にはヒットしたが、ステップ907、ステップ908のいずれへの遷移条件も満たさない場合に遷移する状態である。次のサイクルのプリフェッチアドレスとしては図1のPAG101から出力された信号105の値を用いる。具体的には、図1のセクタ109により信号105を選択し、その値を信号110に出力する。

【0075】ステップ910は命令キャッシュIC104の読み出しが失敗した場合に遷移する状態であり、外部メモリアクセスが起動される。アクセスに用いるアドレスは命令キャッシュの読み出しに用いたプリフェッチアドレス（信号110）である。このアドレス信号線は図1中では省略している。外部メモリからの命令転送が終了すると、状態は再びステップ902へと遷移する。

【0076】ステップ907、908、909の次状態は、ステップ911である。

【0077】ステップ911では、BW102、RB103への書き込み動作を起動するか否かを決定する。その判定は基本的には、無条件分岐命令bra、サブルーチンコール命令bsr、リターン命令rtsのいずれかの命令が命令デコーダ113でデコードされたという、デコード情報を基に起動する。但しこれら3つの分岐命令ごとに起動条件は若干異なる。

【0078】（1）無条件分岐命令bra  
bra命令がデコードされた場合にはBW102、RB103が使用可能状態にある時には常に、BW102への書き込みを起動する。bra命令がBW102を用いた分岐を生じる場合、bra命令は命令プリフェッチ部内で削除され、命令デコード部へは転送されない。逆に、bra命令がデコードされた場合には、そのbra命令はBW102を用いた分岐処理を実行していない。

【0079】（2）サブルーチンコール命令bsr、リターンrts命令

bsr命令またはrts命令を、BW102を用いて分岐処理する場合、命令プリフェッチ部は分岐命令を信号112を通して命令デコード部へ転送すると共に、その分岐命令がBW102を用いて分岐処理済みであることを示すタグ情報を命令デコード部に送る。命令デコード部ではこのタグ情報を基に、BW102を用いて分岐済みの分岐命令について、分岐指示信号と、bsr命令ま



たは `rt s` 命令がデコードされたというデコード情報とを命令プリフェッチ部に送らないようにする。

【0080】BW102への書き込みを起動する場合にはステップ912へ、起動しない場合にはステップ901へ遷移する。

【0081】ステップ912は分岐指示信号待ち状態である。分岐指示信号がアサートされたか否かはステップ913で判定する。ステップ912では図1の命令アドレスキューPCQ121を用いて分岐命令のアドレスを信号線122に出力し、さらに図7の信号線204、205を生成して、分岐指示信号のアサートを待つ。分岐指示信号アサート時にはステップ914へ遷移する。

【0082】ステップ914ではBW102への登録を行う。図1の本実施例のマイクロプロセッサでは、分岐指示信号と分岐先アドレスは同時に生成される。分岐先アドレスは図1のALU117で生成され、信号線118を通してBWに転送される。BW102では分岐指示信号のアサートにより、すべての入力データが揃い、図7のBWA202とBWD203のプリフェッチアドレス110で選択されたエントリに同時に書き込みを行う。本アルゴリズムでは、ステップ914でBW102の書き込みとIC104の読み出しを同時に行う。そのため、BW102の書き込み時にはBW102の読み出しを行えない。但しこれは、BW102を2ポートメモリとすれば、実現できる。

【0083】ステップ914でBW102への書き込みが終了した後は、再びステップ903に遷移する。

【0084】

【発明の効果】以上に示したように、本発明を用いた実施例においては、図4から図6に示したように分岐処理特に無条件分岐処理を高速化できる。その特徴は、

1) 分岐命令アドレスと分岐先アドレスを組にした分岐の履歴情報を保持し、その履歴情報をプリフェッチアドレスを用いて検索するため、早い時点での分岐が可能である。

【0085】2) BW(分岐履歴情報を保持するバッファ)とRB(リターンアドレスを保持するバッファ)を連動して動作させることにより、リターン命令の分岐処理をも高速化できる。

【0086】3) BW内に分岐命令のタイプ情報を保持することにより、上記2)の連動動作が可能になると共

に、分岐命令ごとに極め細かい制御が可能になる。例えば、無条件分岐命令 `bra` の場合には実行処理を削除し、命令 `bsr` の場合には削除しないといった処理の制御にこの情報を使用している。

【図面の簡単な説明】

【図1】本発明の実施例によるマイクロプロセッサの全体構成を示す図である。

【図2】分岐が無い時の図1のマイクロプロセッサのパイプラインの流れを示す図である。

【図3】図1のマイクロプロセッサの分岐命令バッファBW102のミス時のパイプラインの流れを示す図である。

【図4】図1のマイクロプロセッサの分岐命令バッファBW102のヒット時のパイプラインの流れを示す図である。

【図5】図1のマイクロプロセッサの分岐命令バッファBW102のヒット時のパイプラインの流れを示す図である。

【図6】図1のマイクロプロセッサの分岐命令バッファBW102、リターンバッファRB103のヒット時のパイプラインの流れを示す図である。

【図7】図1のマイクロプロセッサ中の分岐命令バッファBW102の構成をより詳細に示す図である。

【図8】図1のマイクロプロセッサ中のリターンバッファRB103の構成をより詳細に示す図である。

【図9】図1のマイクロプロセッサの命令プリフェッチ部の制御フローを示す図である。

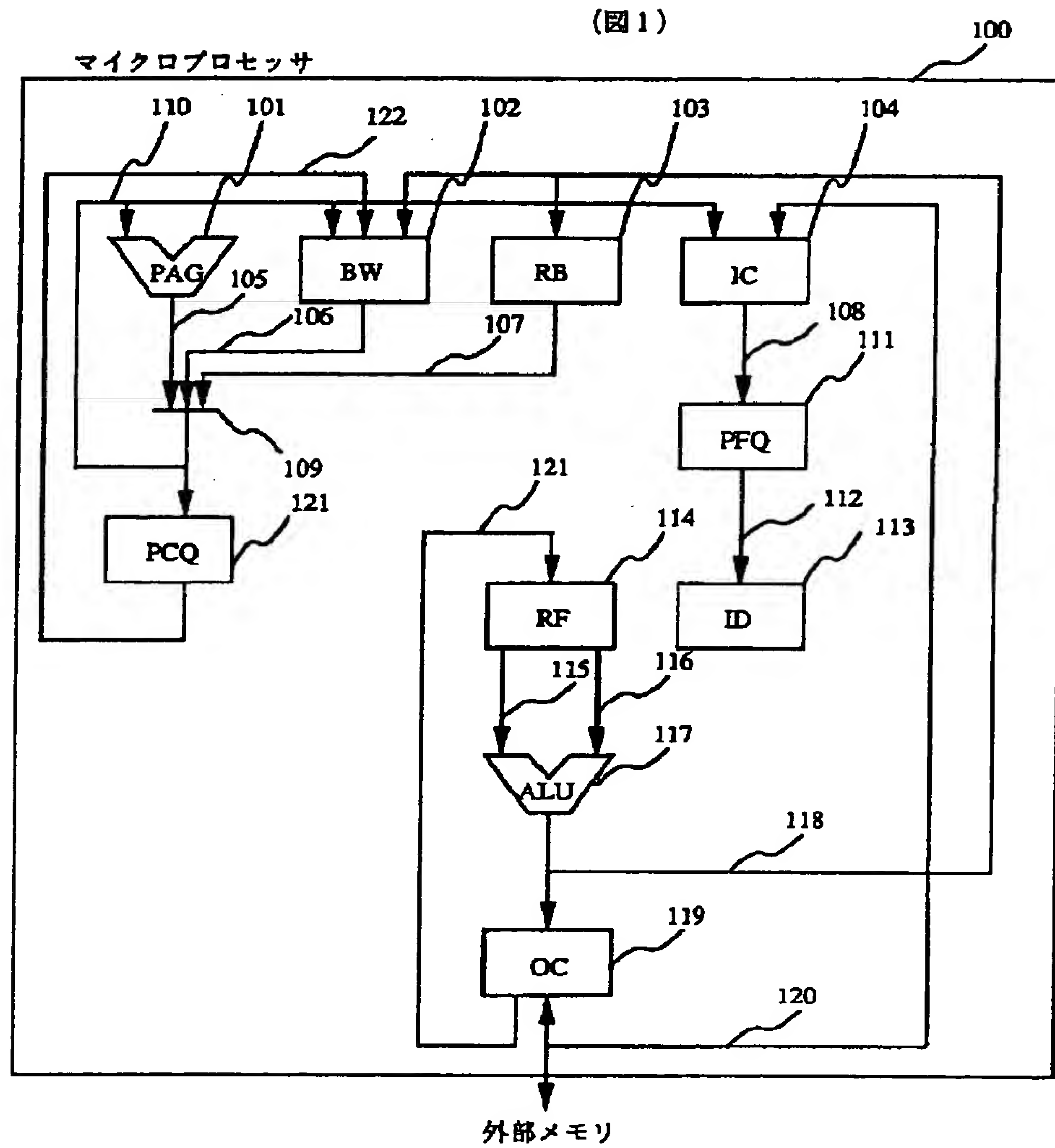
【図10】図1のマイクロプロセッサの命令プリフェッチ部の制御フローを示す図である(図9の続き)。

【符号の説明】

100…マイクロプロセッサ、101…プリフェッチアドレス発生器。29ビット加算器。102…分岐命令用バッファ。分岐先アドレスを保持する。103…リターンバッファ。リターンアドレスを保持する。104…命令キャッシュ。201…5ビットのアドレスデコーダ。202…分岐命令バッファのアドレスタグ部。32エントリ、32ビット幅。203…分岐命令バッファのデータ部。32エントリ、33ビット幅。209…24ビット一致比較器。302…4ビットのアドレスデコーダ。303…リターンバッファのデータ部。16エントリ、32ビット幅。



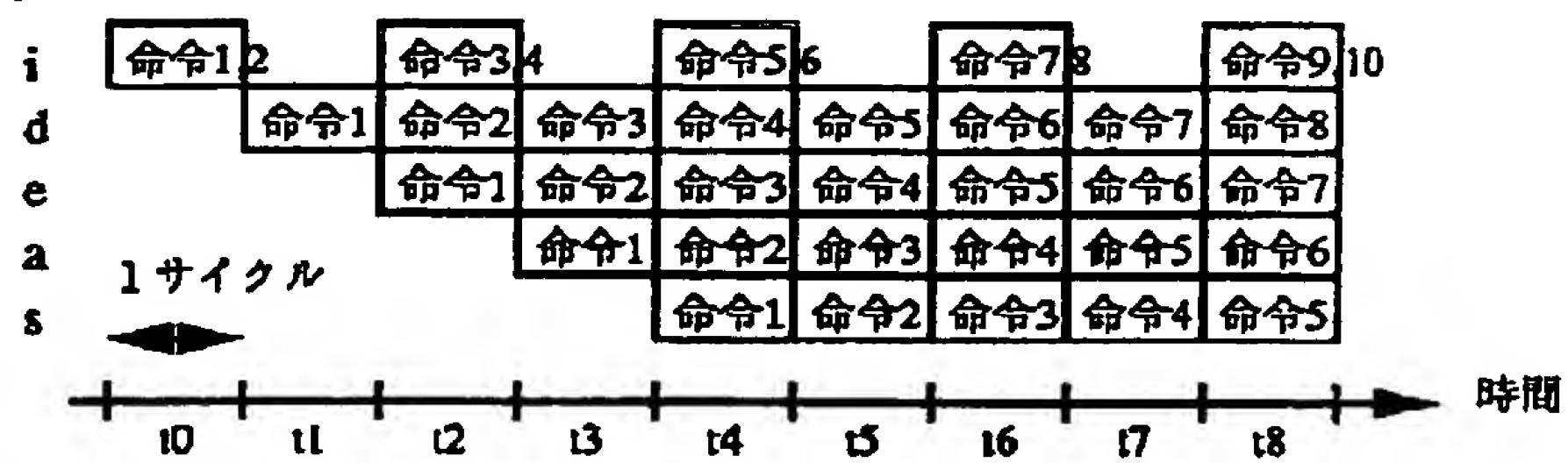
【図1】



【図2】

パイプライン  
ステージ

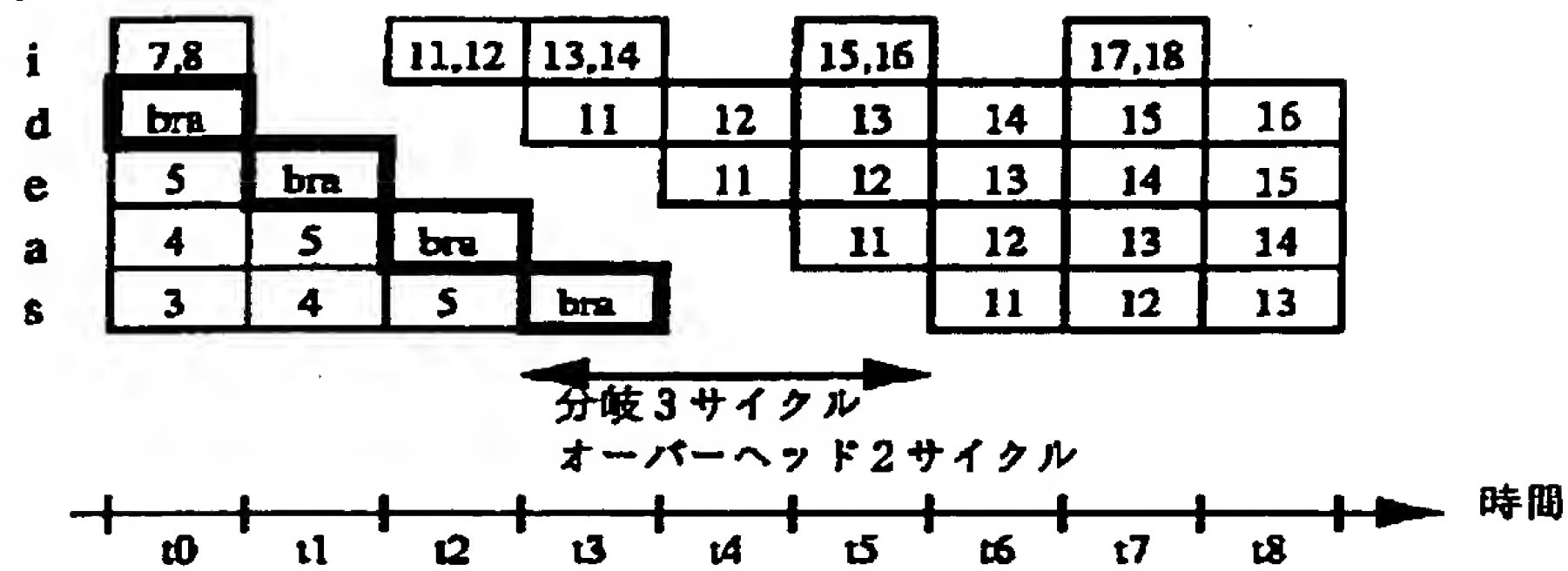
(図2)



【図3】

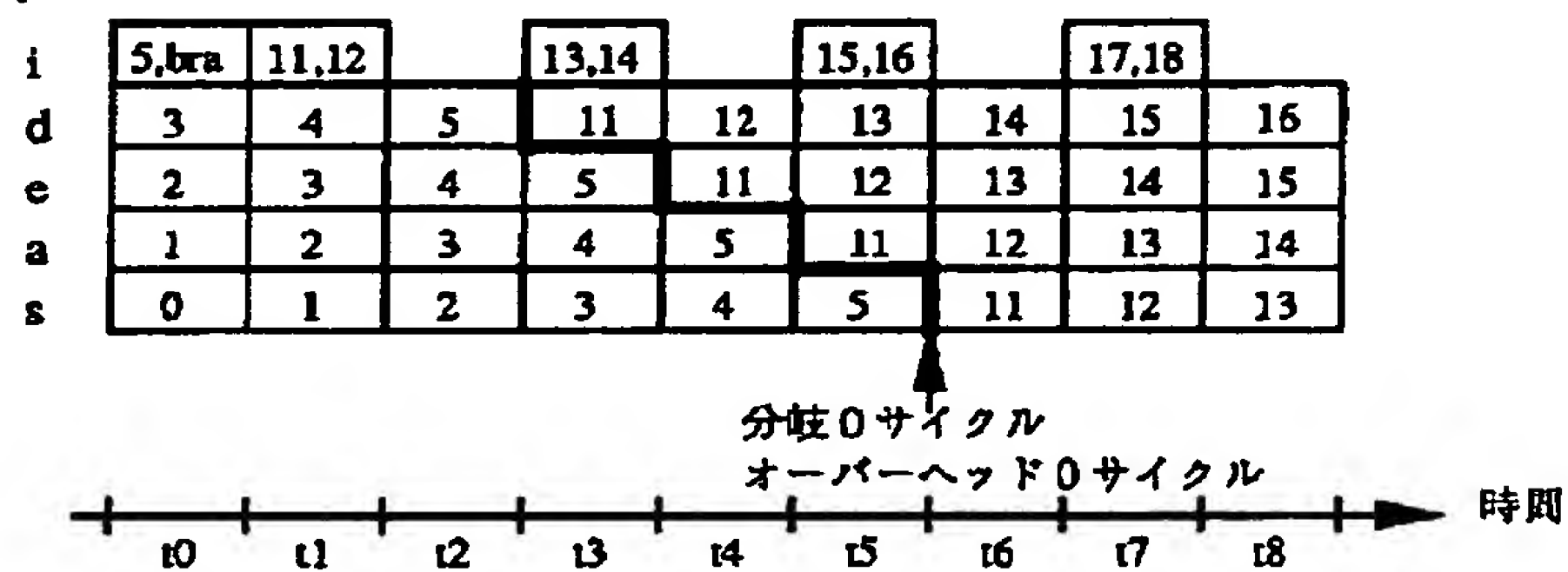
パイプライン  
ステージ

(図3)



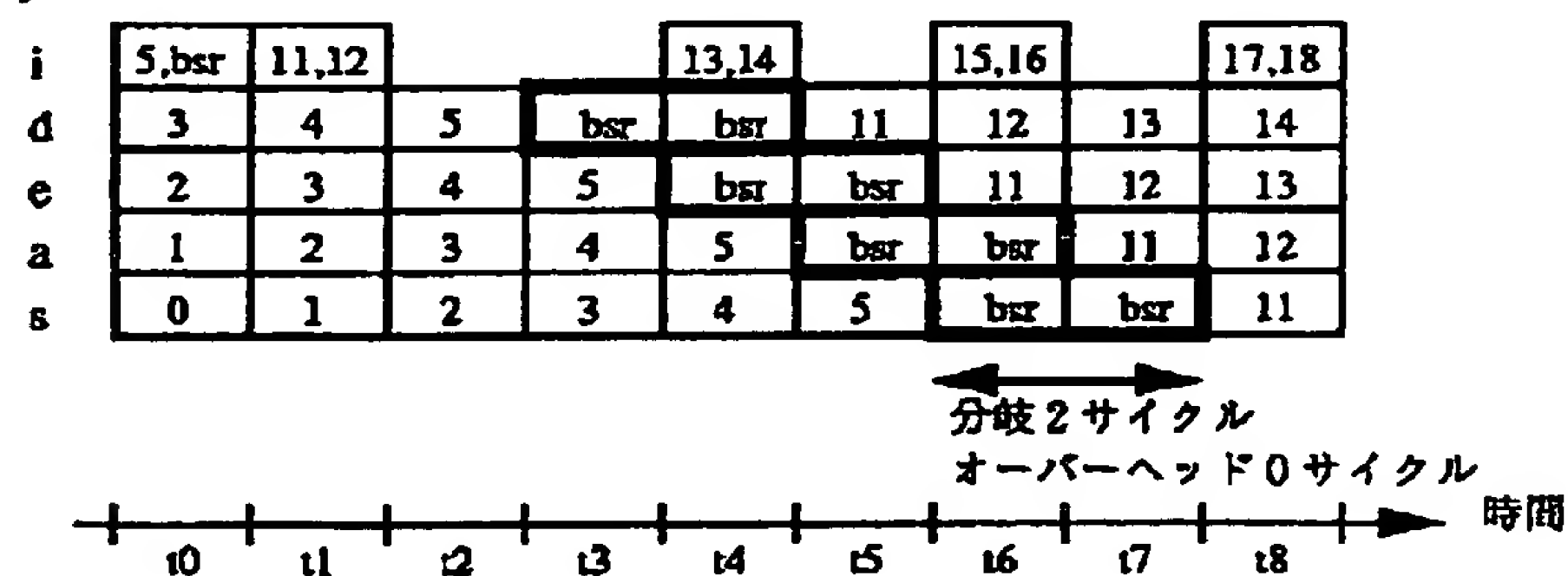
【図4】

(図4)

パイプライン  
ステージ

【図5】

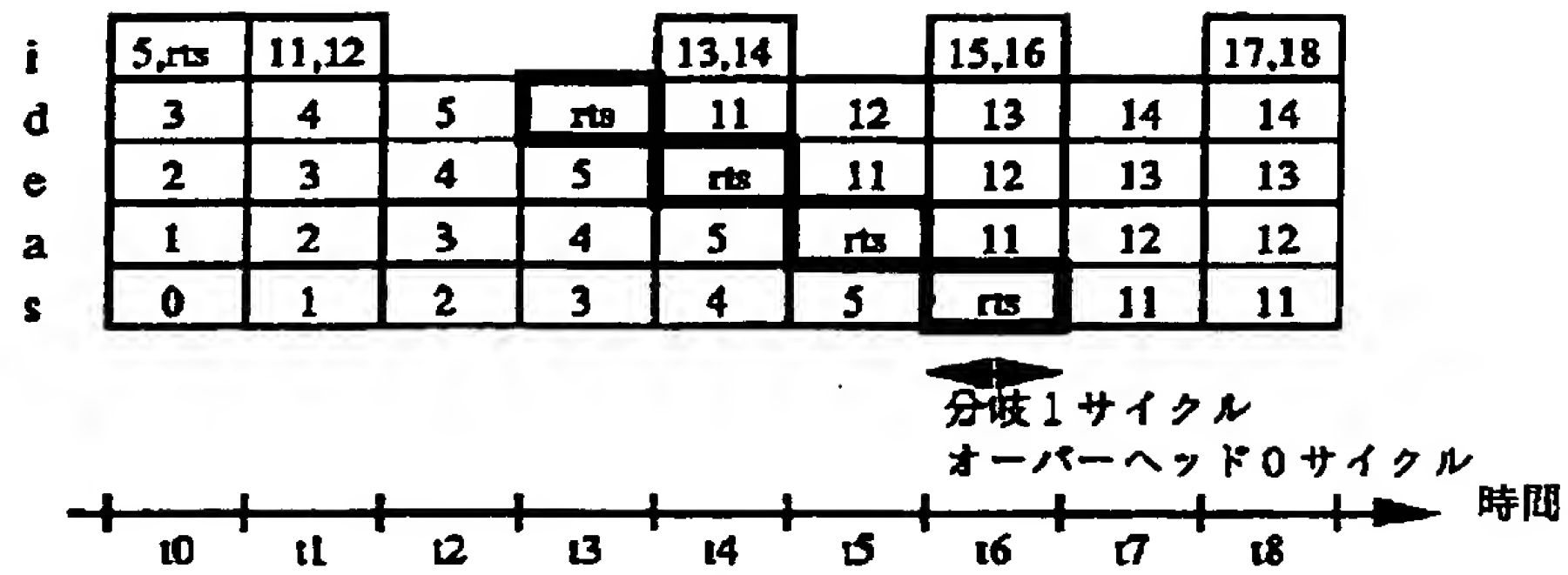
(図5)

パイプライン  
ステージ

【図6】

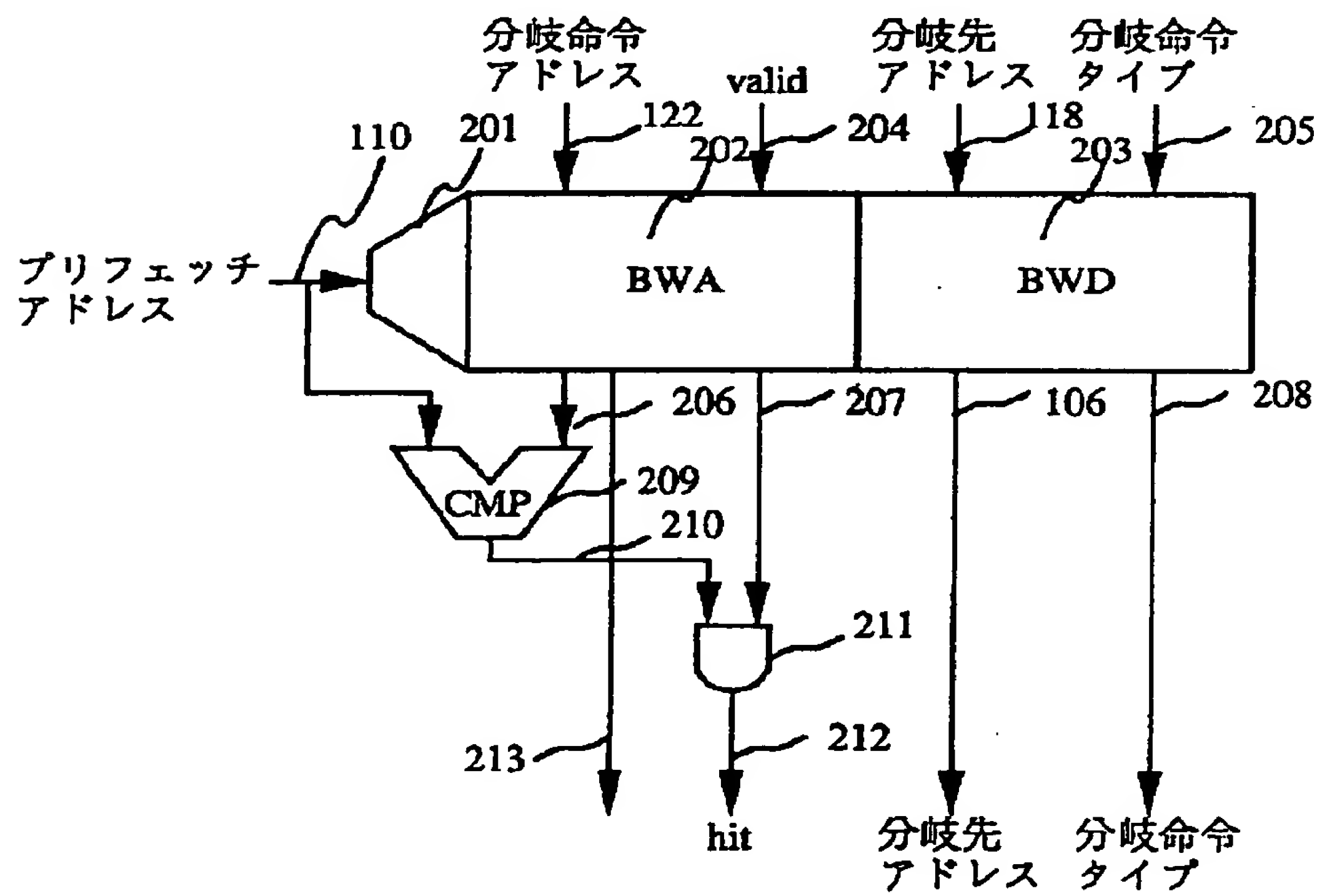
パイプライン  
ステージ

(図6)



【図7】

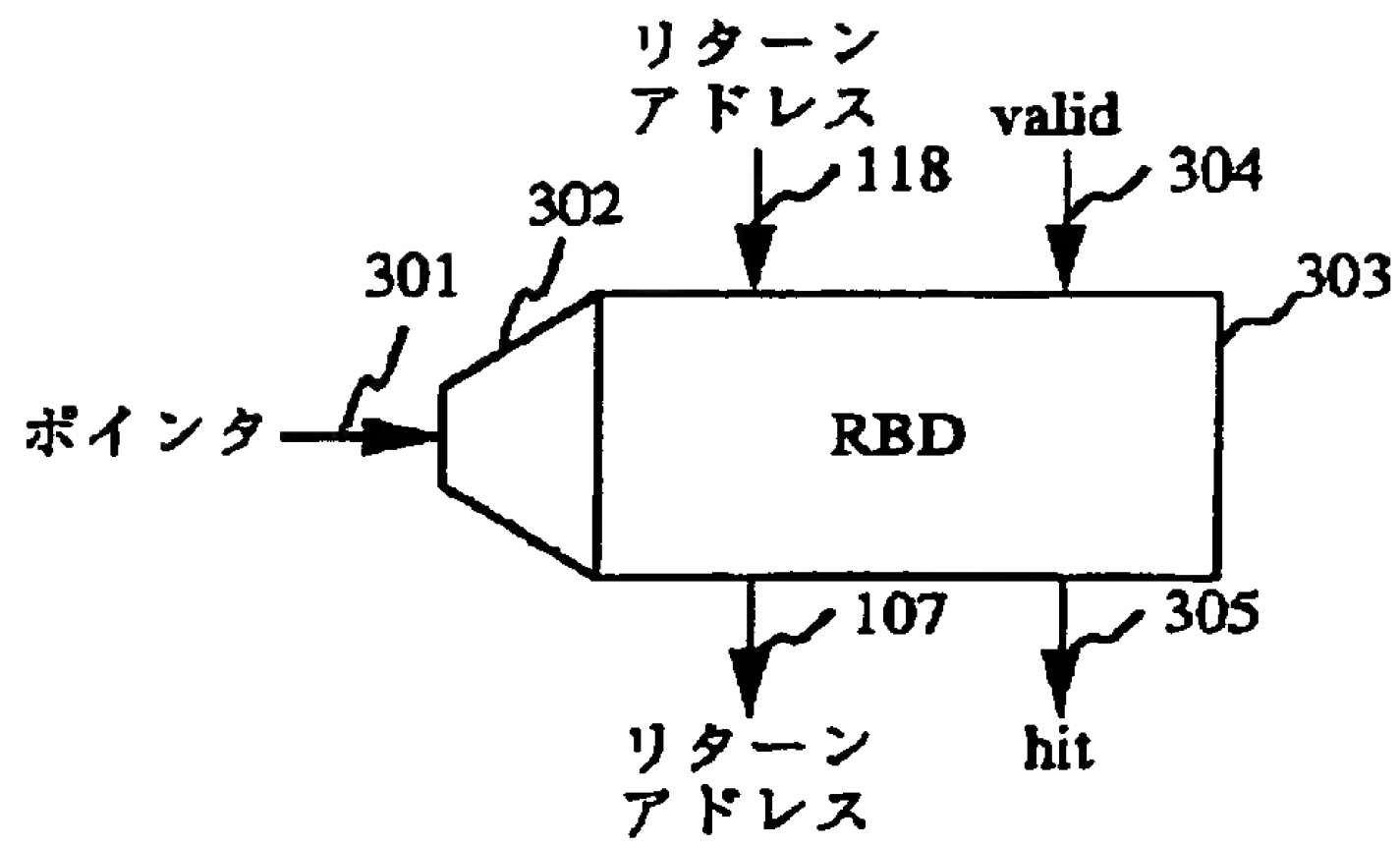
(図7)





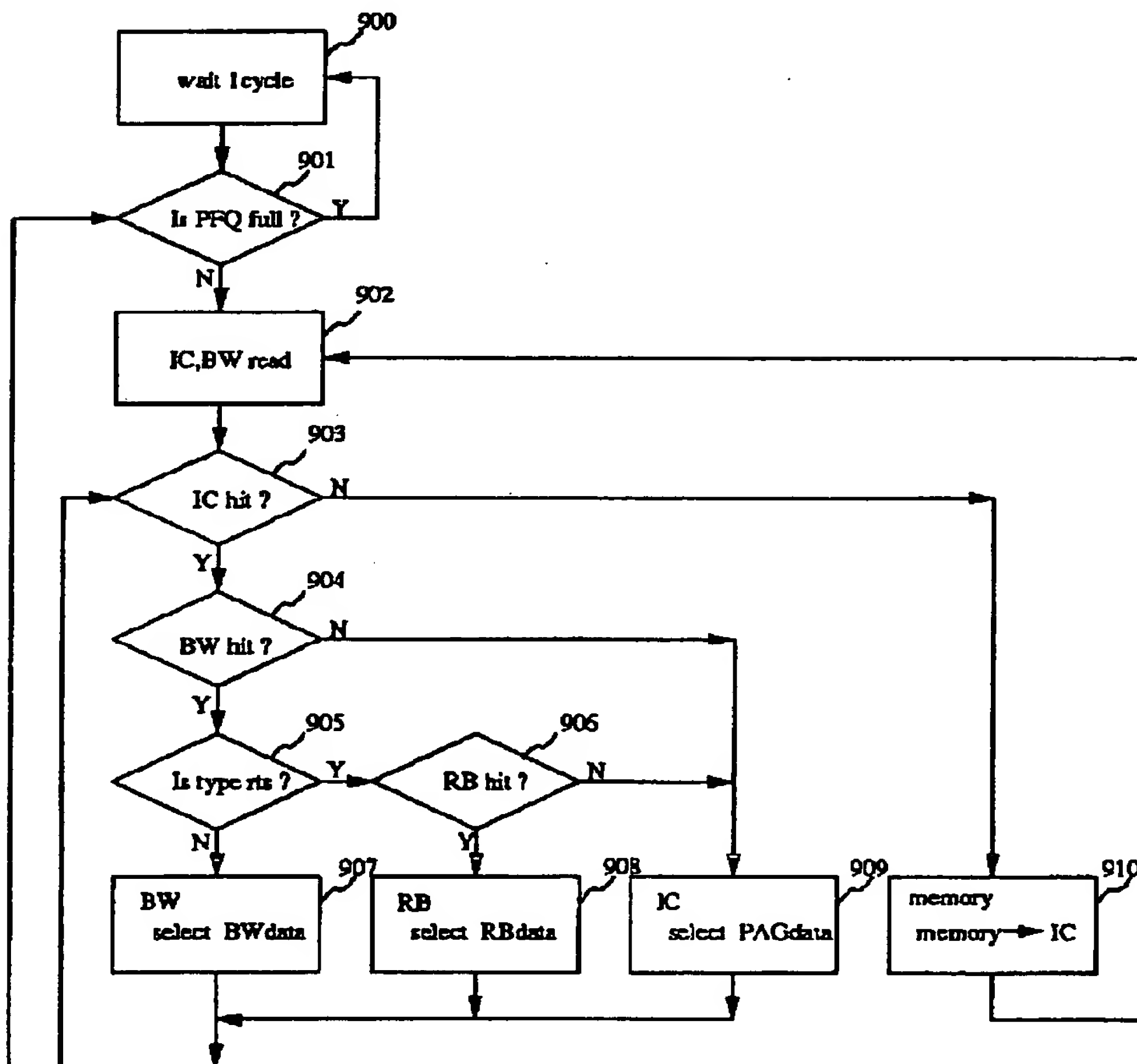
【図8】

(図8)



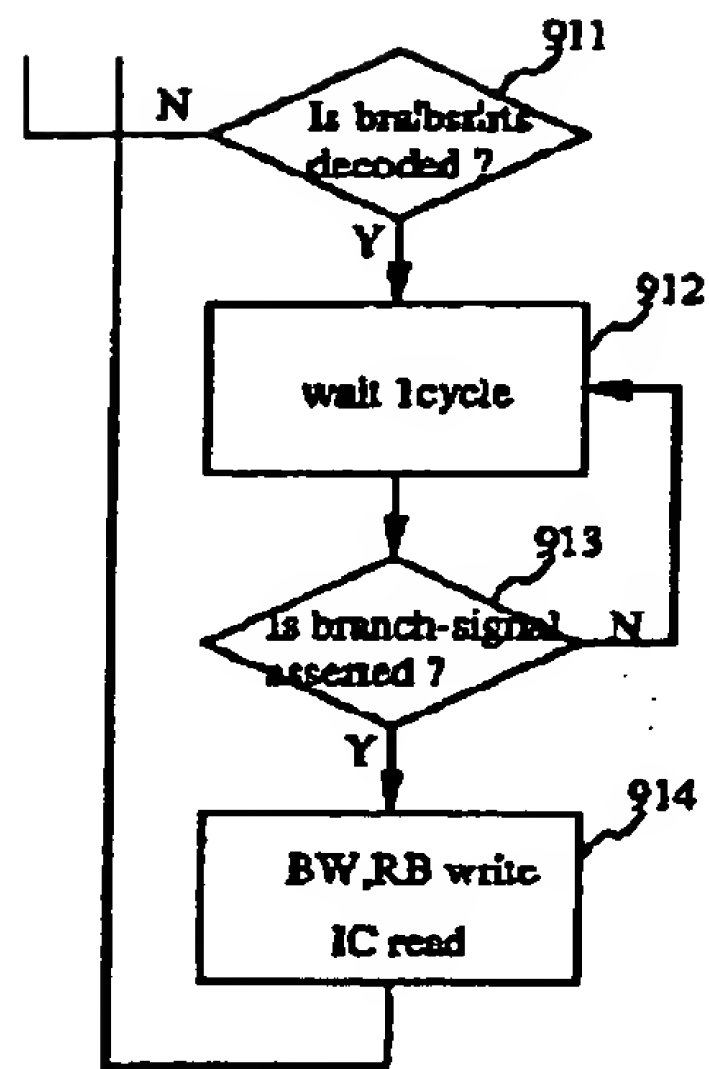
【図9】

(図9)



【図10】

(図10)



---

フロントページの続き

(72) 発明者 青木 郭和  
東京都国分寺市東恋ヶ窪1丁目280番地  
株式会社日立製作所中央研究所内